

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

What We Claim Is:

1. (currently amended) A method for processing data for a system model including the steps of:

providing a model specification having a plurality of types of items including at least one first item type wherein first associated data is obtained from data input into the system and at least one second item type wherein second associated data is obtained from an operation performed on first data associated with at least one third item type stored in a first database,

inputting second data into the system,

automatically searching the ~~input~~ second data for said at least one first item type,

automatically storing ~~said~~ second data associated with said at least one first item type in the first database,

automatically reading ~~the or one of the~~ a second item type from the at least one second item types in a determining step including determining whether the first database includes ~~the or each~~ a prerequisite item necessary to determine the ~~one~~ second item type by obtaining third associated data from an operation performed on third data associated with at least one ~~third~~ fourth item stored in the first data base,

automatically storing the ~~one~~ second item type in the first database if the ~~or each~~ prerequisite item is present;

for each remaining second item type in the at least one second item type, performing the steps of automatically reading and automatically storing said each remaining second item type in the first database if the prerequisite item is present ~~successively reading each other second item type,~~

~~storing said each other second item type in the first database if the prerequisite item is present in the first database, and~~

automatically outputting an indication that the system model can be produced if items of the model specification are stored into the first database.

2. (previously presented) The method as claimed in claim 1 wherein each said second item

type is read successively.

3. (previously presented) The method as claimed in claim 1 including at least two items of the second type.

4. (previously presented) The method as claimed in 1 further comprising: incorporating an iterative process of reading said second item types not stored in the first database whenever said second item type is stored in the first database.

5. (previously presented) The method as claimed in claim 1 wherein said first database further comprises modules; and, said method further comprising the step of storing said first item types in said modules.

6. (previously presented) The method as claimed in claim 5 further comprising: configuring each said module to perform operations on said data associated with said first item types having at least one similar characteristic which are stored in a same said module.

7. (previously presented) The method as claimed in claim 1 further comprising the step of sorting said plurality of types of items and said associated data as said plurality of types of items and said associated data are stored in the first database.

8. (previously presented) The method as claimed in claim 1 wherein said plurality of types of items further comprises predetermined items; and, said method further comprising: the system producing an output indication if said predetermined items are stored in the first database.

9. (previously presented) The method as claimed in claim 1 further comprising the step of determining whether a second item type from said at least one second item type can be stored in the first database by associating the second item type with an item determinant which specifies the or each prerequisite item for evaluation of the second item type.

10. (previously presented) The method as claimed in claim 9 further comprising a determinant step of searching the first database for the or each prerequisite item of the second item type.

11. (original) The method as claimed in claim 10 wherein the determining step includes a Boolean operation which produces a true or false result depending on whether the or each prerequisite item is located in the first database.

12. (original) The method as claimed in claim 11 wherein the first database includes one or more separate storage areas.

13. (previously presented) The method as claimed in claim 12 wherein the result of said determining step is true if prerequisite items are located in the first database.

14. (previously presented) The method as claimed in claim 1 wherein said plurality of types

of items further comprises input items and the first item types correspond to said input items.

15. (original) The method as claimed in claim 1 wherein the second item types have corresponding item determinants.

16. (previously presented) The method as claimed in claim 1 wherein said plurality of types of items further comprises non-input items and the second item types are said non-input items.

17. (previously presented) The method as claimed in claim 14 further comprising the step of adding a second item type from said at least one second item type to the first database if the associated item determinant evaluates to true.

18. (previously presented) The method as claimed in claim 17 wherein said plurality of types of items includes fourth items; and, the method further comprising the step of providing a consolidated storage array for storing said fourth items and for evaluating said item determinants.

19. (previously presented) The method as claimed in claim 18 further comprising the step of evaluating the item determinant for each said second item type not stored in the first database.

20. (previously presented) The method as claimed in claim 19 further comprising the step of storing in the first database each said second item type for which the item determinant is true.

21. (previously presented) The method as claimed in claim 20 further comprising the step of storing said second item types in a second database if associated prerequisite items for said second item types are not located in the first database.

22. (previously presented) The method as claimed in claim 21 further comprising the step of repeating the evaluating step for any said second item type in the second database.

23. (previously presented) The method as claimed in claim 22 further comprising the step of repeating the storage step for each said second item type stored in the second database.

24. (previously presented) The method as claimed in claim 23 wherein the evaluating and storing steps are repeated until the storage step results in no additional said second item types being added to the first database.

25. (previously presented) The method as claimed in claim 23 further comprising repeating the evaluating and storing steps until all said evaluated item determinants are false.

26. (original) The method as claimed in claim 23 wherein the second database comprises a consolidated instance array.

27. (previously presented) The method as claimed in claim 26 further comprising the step of adding said second items for which the item instances evaluate to false to the second database.

28. (previously presented) The method as claimed in claim 27 wherein any said second item

added to the first database after the evaluating step is performed on the second database results in the removal of said added second item from the second database.

29. (previously presented) The method as claimed in claim 28 wherein the evaluation step is repeated on said second item types remaining in the second database if the second item type is transferred to the first database.

30. (previously presented) The method as claimed in claim 29 further comprising the step of storing formula for said second item types in a formula database and evaluating each said first and/or second item type stored in the first database in accordance with an associated formula stored in a formula database.

31. (previously presented) The method as claimed in claim 30 further comprising the step of associating with each said second item type all of said first item types and/or said second item types required before the second item type can be evaluated.

32. (cancelled)